Electronic Fiducialization of Insertion Device Photon Beam Position Monitors

OAG-TN-2004-033

Glenn Decker, September 20, 2004
Accelerator Operations Division, Advanced Photon Source

Recent experience with insertion device (ID) photon beam position monitors (xbpms) has suggested a unique alignment for each monitor, with the side benefit of suggesting a procedure for the reduction of ID gap-dependent systematic errors to better than +- 10 microns.

In an attempt to understand why the different xbpms have wildly varying dependence on insertion device gap, I decided to log the individual blade signals while running the script SRCollectGapFFData. This script is used to scan a large number of ID device gaps and generate the lookup tables used to compensate for idxbpm gap-dependent systematic position errors. By plotting the data in a number of different ways, one can clearly see that the fundamental differences in the shapes of the curves, whatever the cause, will result in large erroneous gap dependence.

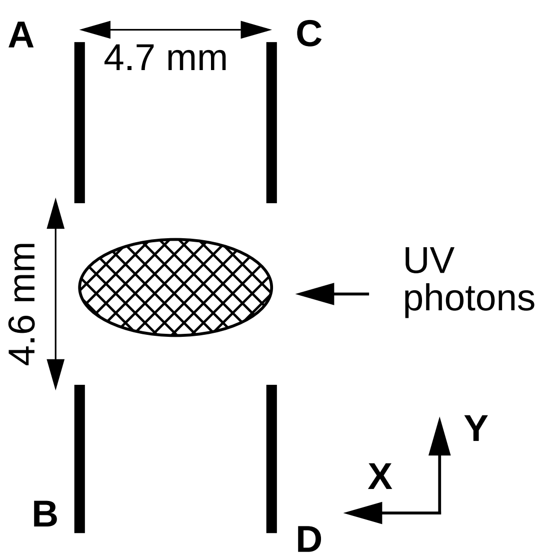


Figure 0:
P1 IDxbpm blade geometry

Shown in Figure 1 are three plots of the same four “adjusted” idxbpm blade signals vs. gap, on a linear scale, normalized, and on a semilog scale. The blades for this particular bpm, 12IDP1, use the so-called ABCD geometry:

Here the term “adjusted blade signal” means that the raw signal is first normalized to stored beam current, after which a constant corresponding to the open-gap stray radiation background signal is subtracted..

The beam was badly misaligned for the data shown, as can be seen by the large imbalance in signal levels. Doing this tends to exaggerate the difference in the shapes of the traces, most noticeable as slope differences on the semilog plot at the bottom.

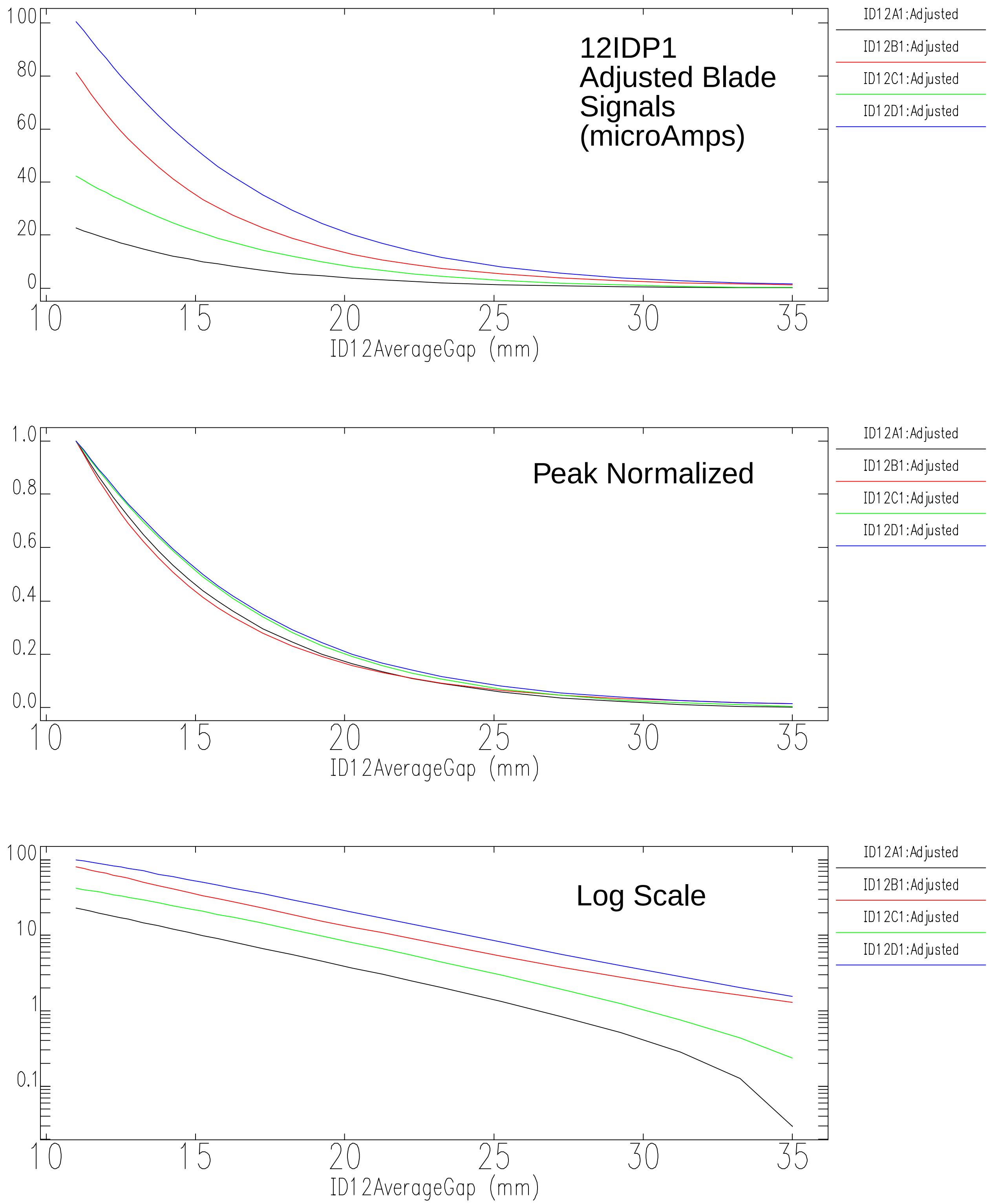


Figure 1 Variation of blade signals with gap for badly mis-aligned beam

Qualitatively, the shapes of the curves seem to match better if the beam is more “centered”, as seen in the analagous plot of the same four signal shown in figure 2.

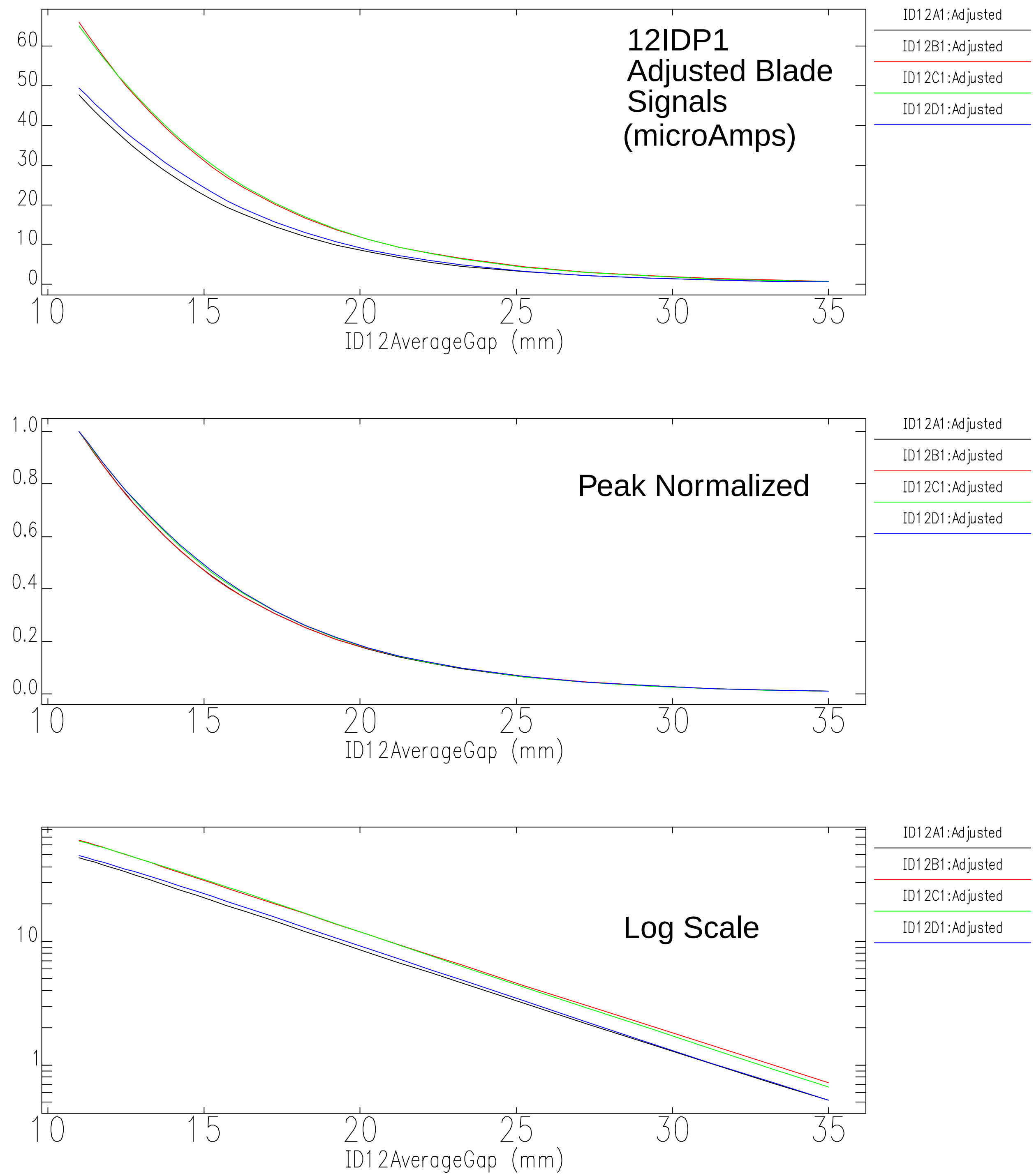


Figure 2. Adjusted blade signal variation with gap for “centered” beam.

Figures 1 and 2 seem to suggest that there is some optimal alignment which makes the blade variation with gap most similar, which should result in reduced erroneous position variation with gap. This should be evident simply by recognizing that the computed positions are simply ratios of differences and sums of individual blade signals.

In order to belabor the obvious, however, I'll write down a couple of equations. Suppose that the curves are all pure exponentials (they aren't) but with different exponent coefficients corresponding to the slopes of each individual blade signal on the semilog plots of figures 1 and 2. I'll use a two-blade bpm as an example (the P2 idxbpm has one top blade labeled A, and one bottom blade B, for example). I'll assume that the constant background subtraction has been done correctly so that

$A = A_0 \exp[- \alpha * \text{gap}]$; $\qquad B = B_0 \exp[- \beta * \text{gap}]$,

where α, β, A_0, and B_0 are constants. The computed vertical position Y is then something like

$$Y = G * (A - B) / (A+B) \qquad (1)$$

where G is a calibration factor. Just doing the math in your head, you can see that the only way to eliminate the gap dependence from Y is to make $\alpha = \beta$. So hopefully the horse is dead by now*.

In order to elucidate more clearly exactly how one might go about performing such an alignment, a series of data was collected for a variety of different steering conditions. Essentially I stole the sddsexperiment input file generated by SRCollectGapFFData, and updated it to log the blade signals in addition to all the data it already collects. A total of 21 insertion device gaps were simultaneously scanned.

Ideally one would like to use the mechanical translation stages to displace the bpms relative to the beam from one scan to the next, however only two translation stage controllers are available. Instead, a "simo-bump" was created. What this amounted to was running controllaw with a perfectly square matrix, using 80 steering correctors and 80 rf beam position monitors. The rf bpm's were chosen in pairs, one set straddling each insertion device straight section. Then, by incrementing the rf bpm setpoints, the same angular change was made simultaneously at all the id source points. Because the matrix is square, the correction was exact, changing the rfbpm

* A reference to the colloquial expression "beating a dead horse"

readbacks by exactly the amounts specified. The upstream rfbpm setpoints were changed in opposition to the downstream setpoint change, to effect essentially a pure “angle bump”. The step size chosen corresponded either to 250 microns of displacement at the P1 xbpm location (16 meters downstream of the center of the id straight) and later reduced to 100 microns as the alignment converged near the optimal steering condition.

Shown in figure 3 is an example of the type of data collected. A total of seven scans were performed in each plane, producing the family of curves shown corresponding to the horizontal plane. An analagous set of curves for the vertical plane is shown in Figure 4. Data for all 42 idxbpm's was collected simultaneously.

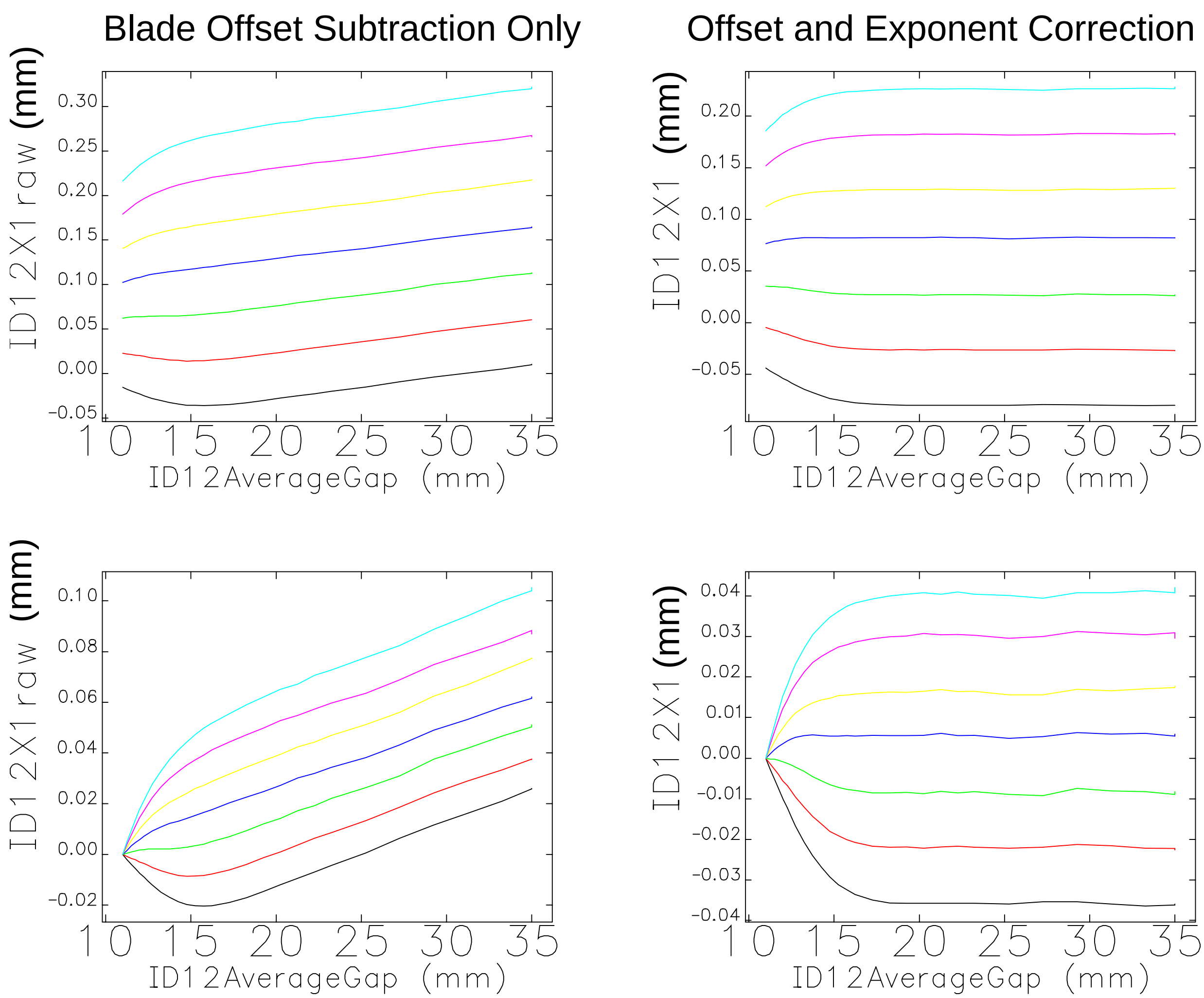


Figure 3. Computed P1 horizontal position variation as a function of gap, for a variety of horizontal steering settings.

All four plots in figure 3 are derived from the same 7 data sets, but with different corrections applied. For each curve, each individual blade signal was fitted using the program sddsexpfit to a function of the form

$$F(gap) = Constant + Factor * Exp[- Rate * gap]$$

The range of gap values fitted were from 19 to 35 mm. It was found that the blade signals follow a pure exponential quite closely in this range. At smaller gaps, there is a lot of evidence suggesting that a cloud of photoelectrons inside the xbpm vacuum housing causes non-exponential behaviour, as well as blade-to-blade crosstalk.

The left hand plots in figure 3 display computed positions in mm, derived from adjusted blade signals with only the fitted constant subtracted. Each curve used different constants, since the background radiation corresponding to the fitted constant has a very strong dependence on steering. For the right hand plots, each blade signal was multiplied by a factor Exp[+Rate] after blade offset subtraction. This second level of correction tends to flatten out the gap dependence for gaps greater than 20 mm.

The bottom pair of plots show the same data as the top pair, but each curve is shifted to a common value at minimum gap. The top pair indicate the absolute range of steering values (hundreds of microns), while the bottom pair accentuate the level of gap variation, generally on the scale of tens of microns.

This isn't quite true because the vertical axis actually uses delta over sum units, i.e. the calibration factor G in equation (1) is set to unity. In this particular case the actual gain is 0.998, but this is generally not the case. The fact that the curve spacing varies with gap is an indication that the calibration factor itself is a function of insertion device gap. One other item to note is that the calibration factors are generally determined at a gap of 15 mm.

The blue trace in the center of the lower right hand plot corresponds very nearly to what can be defined as optimal horizontal steering for this unit. The gap variation after correction for offset and exponent is less than 10 microns. Note that prior to exponent correction (the plot on the lower left) that the variation with gap is greater than 60 microns.

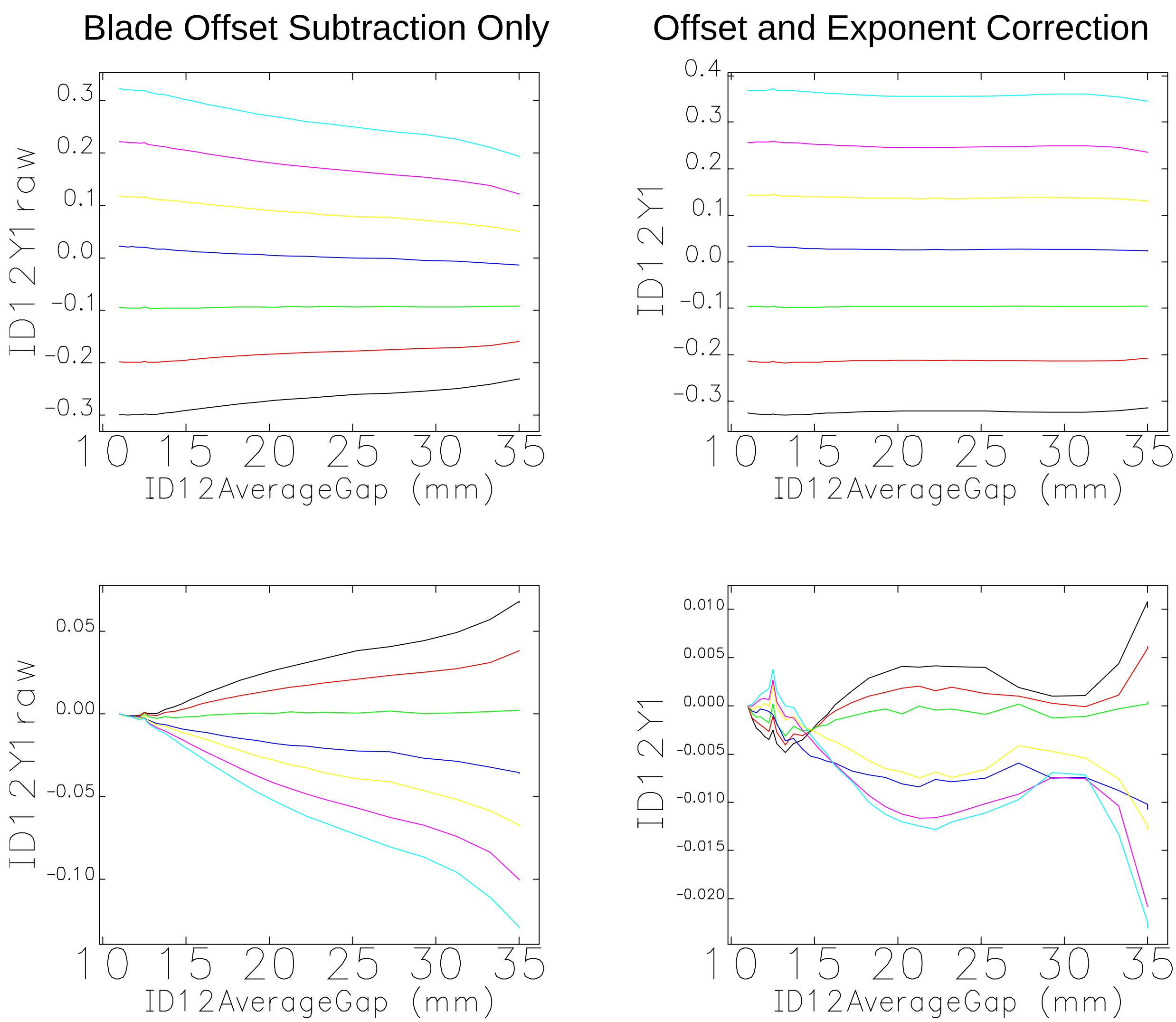


Figure 4. Computed P1 vertical position variation as a function of gap, for a variety of vertical steering settings, in dimensionless delta over sum units.

Shown in figure 4 are analogous data for the vertical plane. It was usually the case vertically that signficant improvement in gap variation could be achieved by careful alignment and offset subtraction alone. The additional improvement from exponent correction helped compensate for misalignment. In figure 4 it is the green curve that corresponds to optimal steering, showing much less than 10 “microns” of variation, with or without exponent correction, once the alignment is correct. The calibration factor is off by about 10% at 15 mm gap in this case. The spacing between curves is “really” 100 microns, independent of gap, at least assuming the rf bpm’s used for the simo-bump are correctly calibrated.

The conclusion is that by careful measurement and compensation for background signal levels a recipe for the determination of optimal idxbpm alignment suggests itself. Namely, for any given unit one should manipulate the mechanical translation stage such that the delta over sum signal lies on an optimal curve, for example the blue trace of figure 3. Referring to the upper left hand plot, one should arrange that delta over sum equals approximately +0.100 at minimum gap (11 mm).

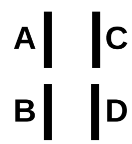
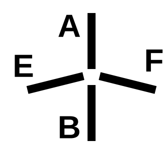
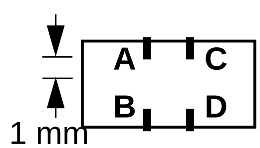
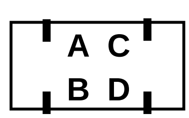
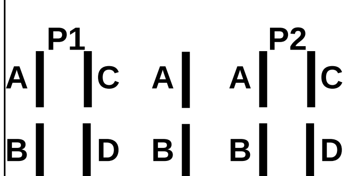
This procedure was followed through two iterations for the majority of insertion device beamlines over the course of run 2004-2. The data shown in figures 3 and 4 were collected following the first iteration. The alignment itself involved moving the two translation stage controllers around the ring as each sector was visited in turn. Once the alignment was complete, a followup data set needed to be collected to validate the alignment.

The control system as it was configured for run 2004-02 allowed for blade offset compensation, meaning that the performance shown in the left hand plots of figures 3 and 4 is realizable in practice. Work will proceed during the next operational period to implement in EPICS the exponent correction needed to achieve the performance shown in the right hand plots. If this effort is successful, it is possible that use of the present ID gap feedforward algorithm can be discontinued altogether.

Influence of xbpm blade geometry

Figures 3 and 4 are typical of performance for the original so-called P1 ABCD geometry sketched in figure zero. It is important to bear in mind that there are several other geometries in use at the different beamlines. The geometry has a profound effect on the level of improvement possible. Table 1 indicates the different blade geometries and their locations, together with a relevant comment. Mechanical drawings showing projections of the different blade geometries are shown in figures 5, 6, and 7.

Table 1: Summary of idxbpm blade geometries and locations

Geometry	Cartoon	Location	Comments
Original P1		Most Upstream P1 locations	Best performance
Original P2		Most Down-stream P2 locations	Strong space charge effects at small gaps. E and F blades have small signal above 25 mm gap.
New P1		Upstream P1 in sectors 16, 22, 31, 32	Small signal above 24 mm gap, very non-linear response. Extremely sensitvie to alignment. Upstream mask limits blade intrusion to 1 mm.
New P2		Downstream P2 in sectors 16,22,31,32	Tiny signal above 24 mm gap, non-linear response. Extremely sensitive to alignment
Canted Undulator		Sectors 21,23,24	Two beams detected by single 10-blade device. P1 / P2 crosstalk at small gaps, from space charge..

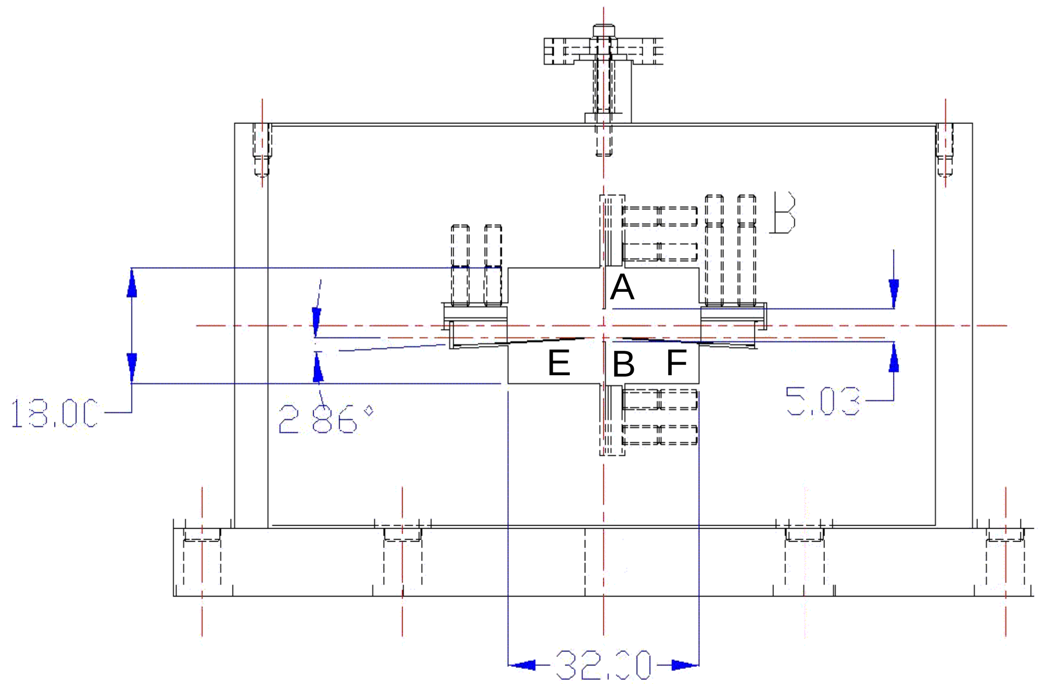


Figure 5. Projection of “original P2” idxbpm blade geometries. Dimensions shown are in mm.

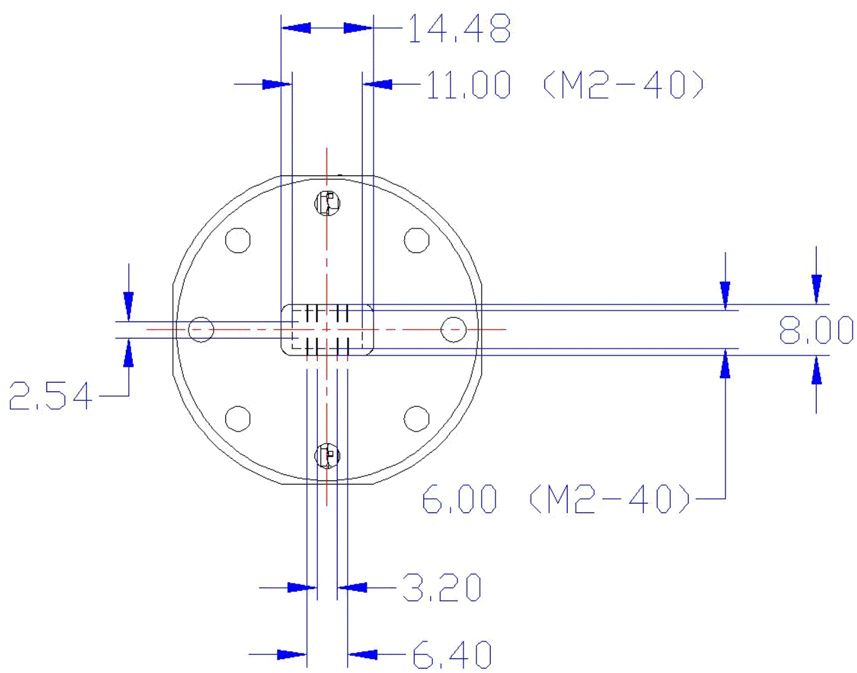


Figure 6. Projection of “new P1” and “new P2” idxbpm blade geometries.

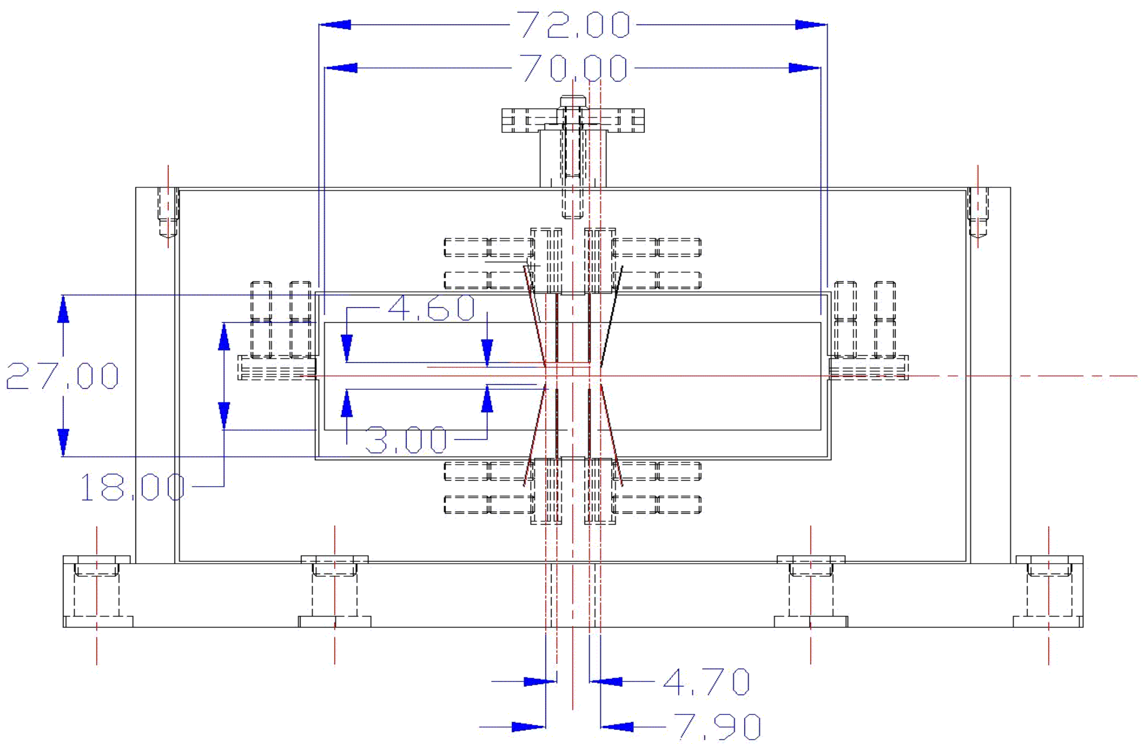


Figure 7. Projection of “original P1” and a proposed replacement for the “original P2” blade geometries.

Shown in Figure 8 is a “typical” family of curves associated with the “Original P2” horizontal readback. In fact there is a lot of variation from unit to unit in the shapes of these curves. Recall that the P2X readback is derived from the E and F blade signals. and that there are large horizontal asymmetries in the background radiation field. Also, the signal level drops more rapidly with gap for E and F than for any of the other blade signals in the original P1 or P2 design, as can be seen from figure 9. This tends to make performance at gaps larger than about 25 mm very unstable.

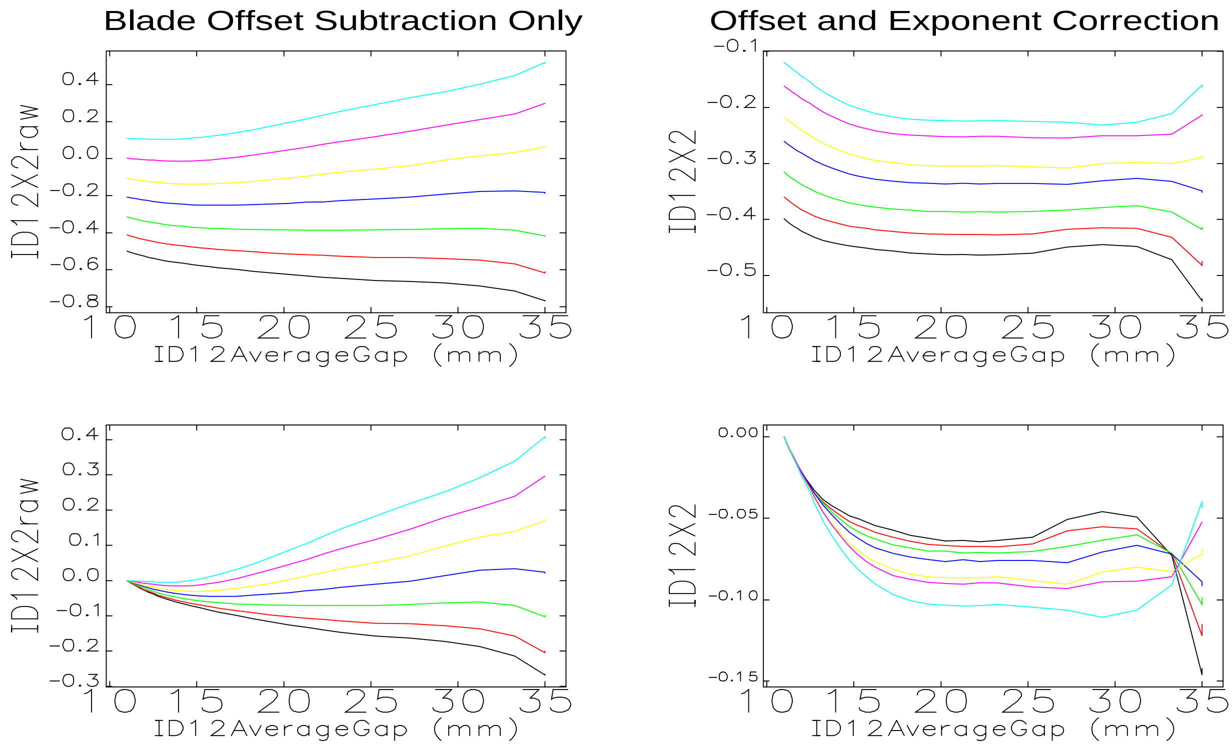


Figure 8 Computed “original P2” horizontal position variation as a function of gap, for a variety of horizontal steering settings.

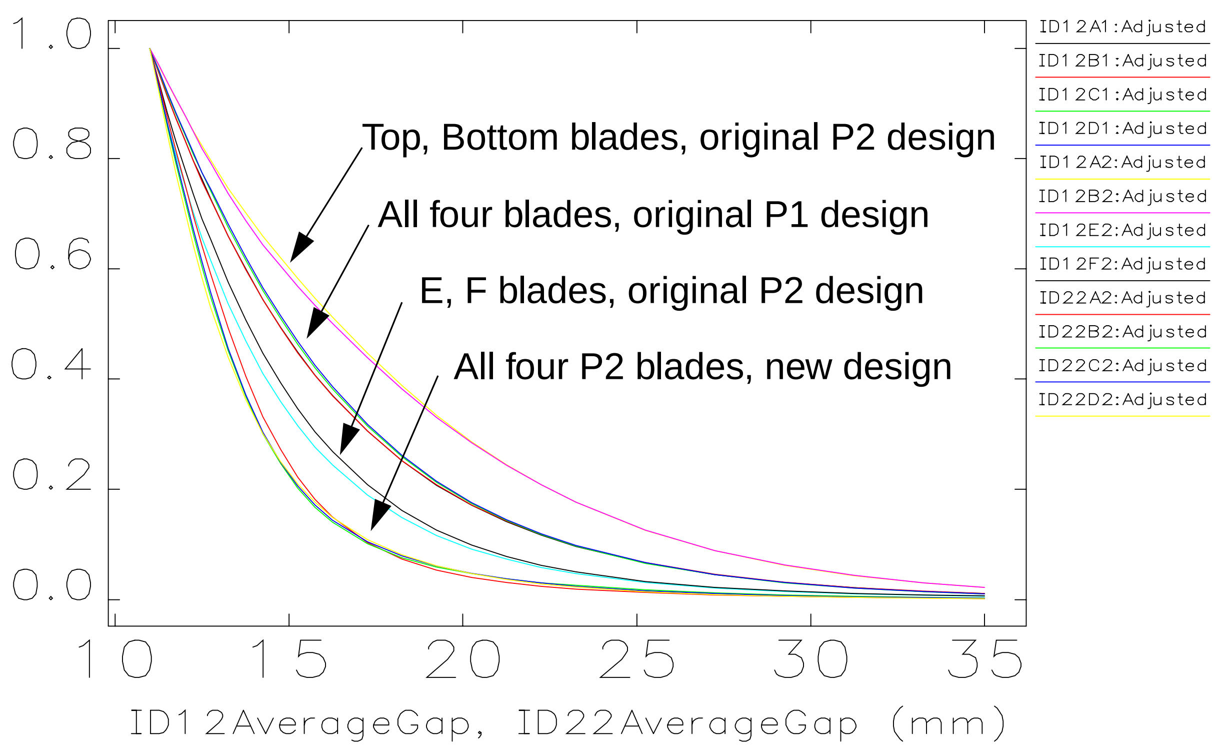


Figure 9 Variation of normalized blade signals with gap for various blade geometries.

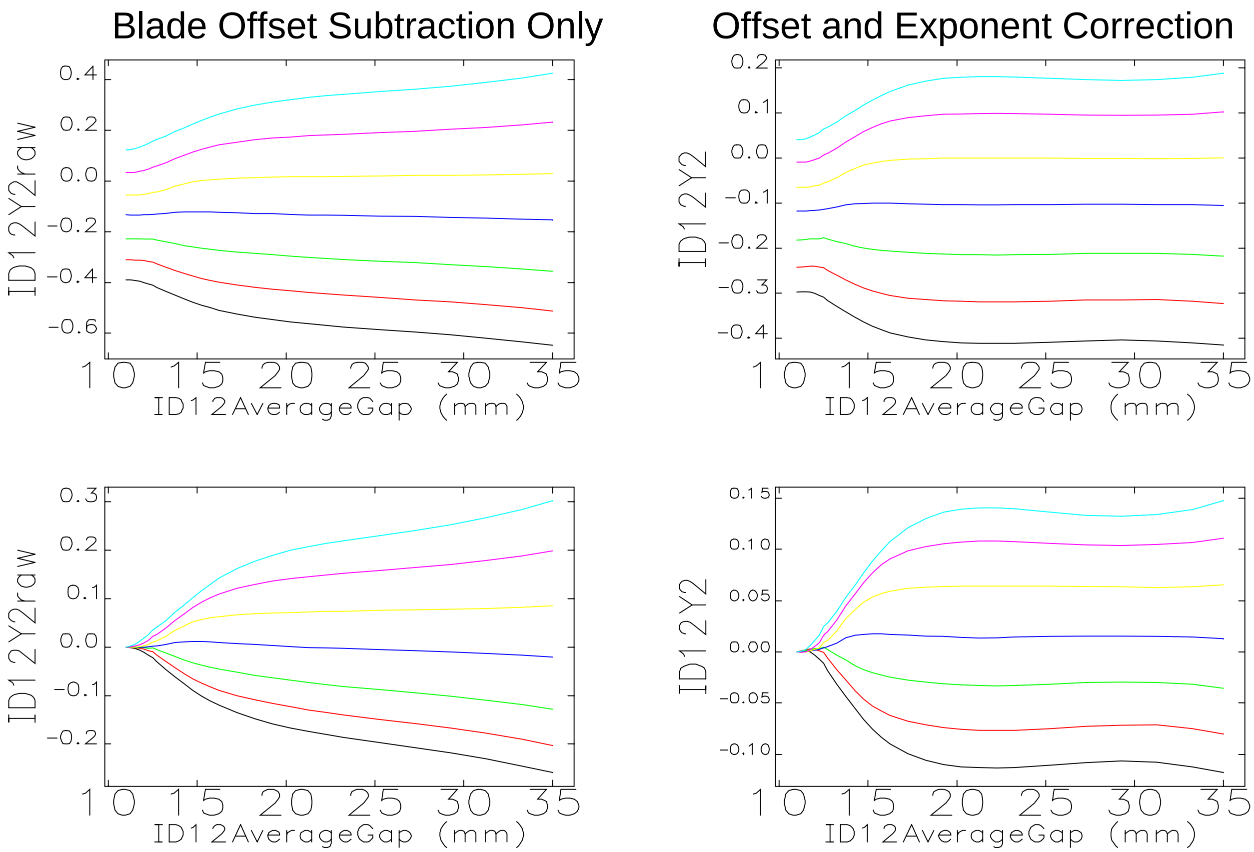


Figure 10 Computed “original P2” vertical position variation as a function of gap, for a variety of vertical steering settings.

In the case of vertical P2 readbacks from the original P2 geometry, the position is calculated from the A and B blade signals only. Shown in Figure 10 is the variation of P2Y with gap. The shapes of these curves tend to be similar from unit to unit. The sharp curvature seen at gap values below 13 mm or so is thought to have something to do with a cloud of photoelectrons generated by the high photon flux,

Notice that the vertical tick marks in the lower right hand plot of figures 8 and 10 are spaced by 0.05 delta over sum units, to be compared with 0.01 and 0.005 units respectively in figures 3 and 4 for the original P1 geometry. A plan to replace the blade assemblies in the P2 vacuum housing is underway, which should improve performance considerably. The angled blades shown in figure 7 are presently under investigation as a viable option.

New ID front ends

Some of the newer beamline front ends, specifically in sectors 16, 22, 31, and 32, were value engineered and have substantially different xbpm geometry, hinted at in table 1 and shown explicitly in figure 6. One of the main differences is that a collimating mask is located upstream of the first xbpm (P1), allowing only 1 mm of the tip of each blade maximum to protrude into the beam. This is represented by the rectangle surrounding the blades in table 1. Given that the translation stage has a range on the scale of a few mm, it is genuinely possible to retract the top or bottom pair completely from the field of view, meaning that alignment for these units is critical.

Another significant difference is that the new front end P1 horizontal blade separation is only 3.2 mm vs. 4.7 mm for the original P1. The vertical separation between the tips of the blades is also significantly smaller, 2.5 mm vs. 4.6 mm for the original P1. In general this means that the position response is more nonlinear for these units. The signal strength in terms of microamps as a function of gap, is comparable to the original P1 design.

The horizontal separation for the new P2 geometry is 6.4 mm, i.e. larger than for the original P1 design. The new P2 vertical blade separation is also 2.5 mm. This results in signal strengths that fall off more rapidly with increasing gap than for any other geometry, as seen in figure 9.

Figures 11-14 show the position variation with gap and steering for the new geometries

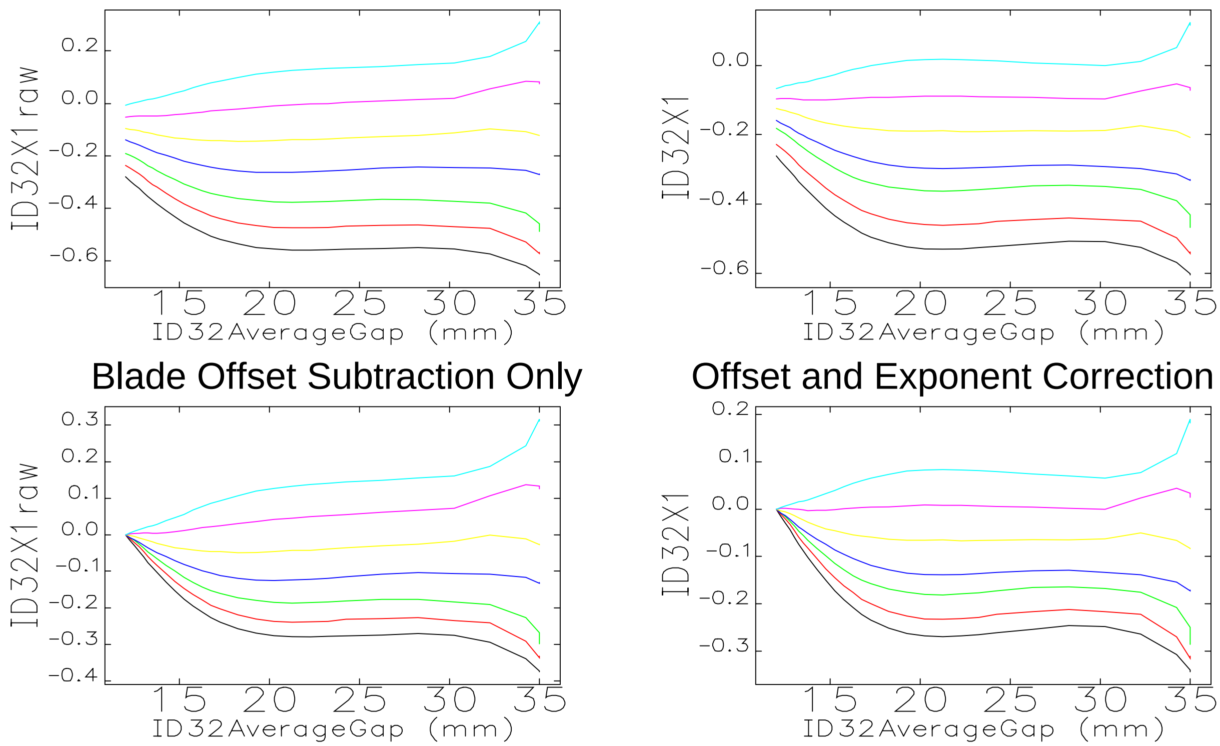


Figure 11 Computed “new P1” horizontal position variation as a function of gap, for a variety of horizontal steering settings.

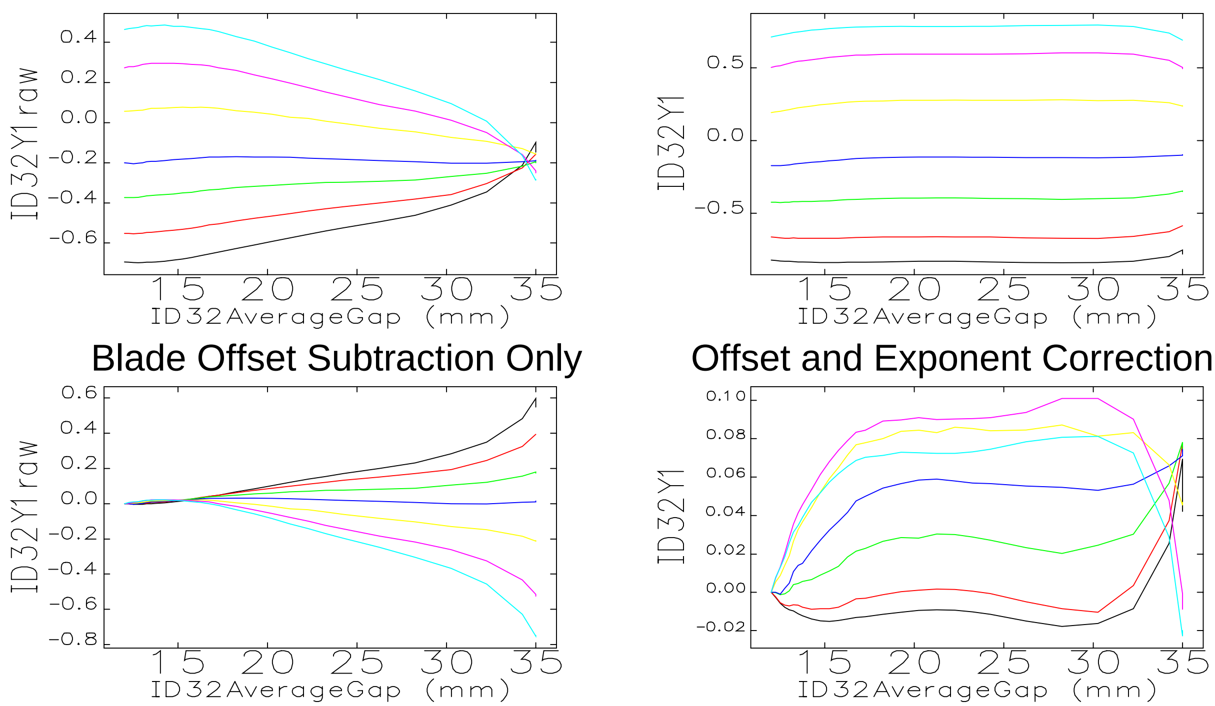
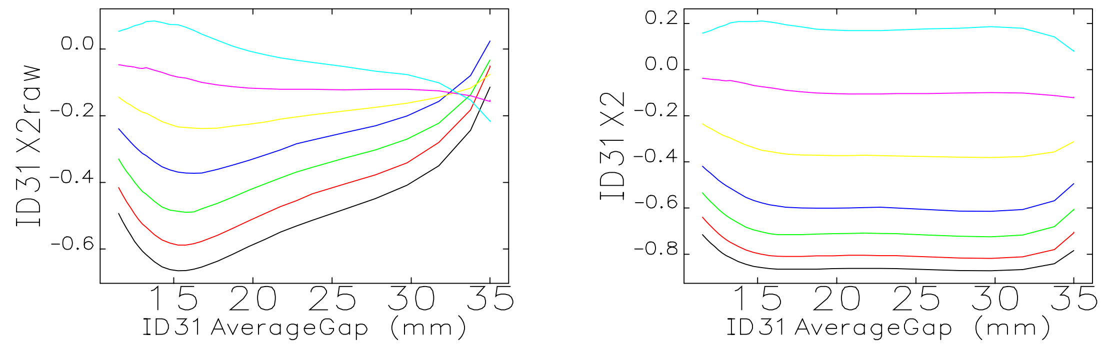


Figure 12 Computed “new P1” vertical position variation as a function of gap, for a variety of vertical steering settings.



Blade Offset Subtraction Only Offset and Exponent Correction

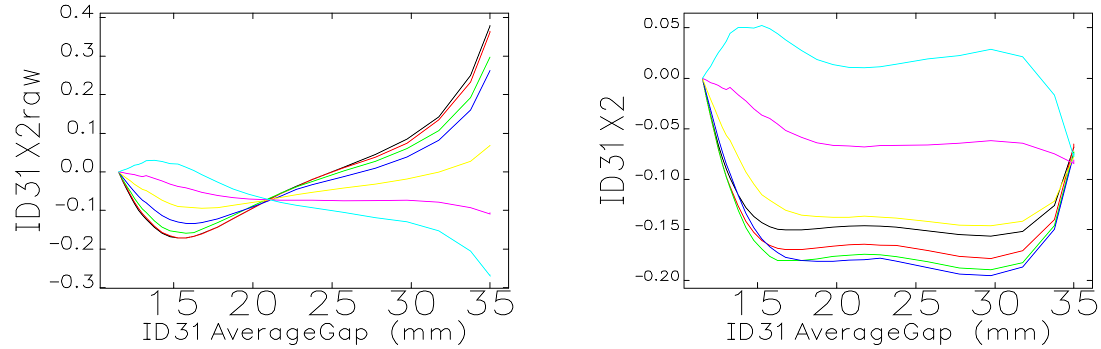
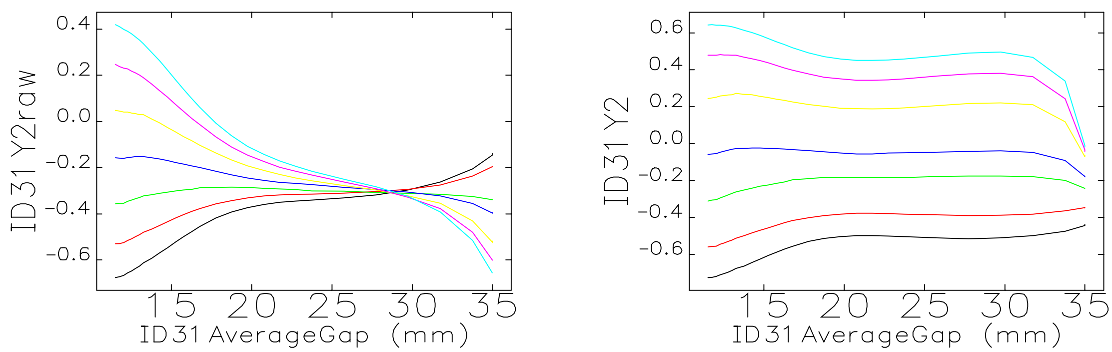


Figure 13 Computed “new P2” horizontal position variation as a function of gap, for a variety of horizontal steering settings.



Blade Offset Subtraction Only Offset and Exponent Correction

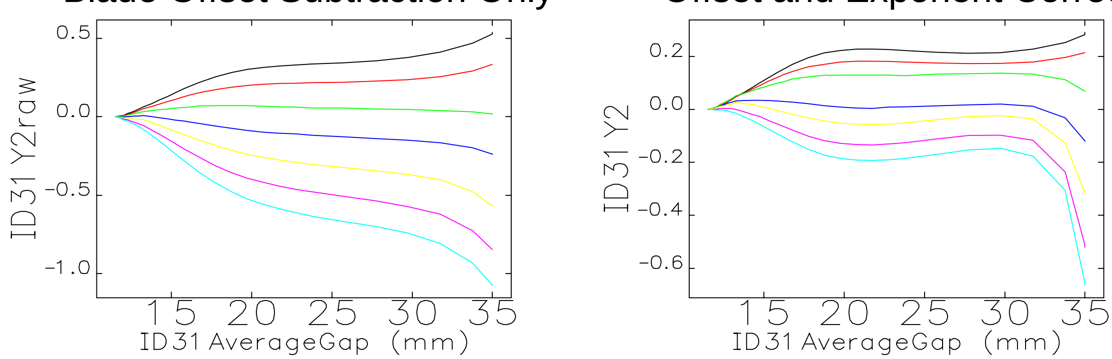


Figure 14 Computed “new P2” vertical position variation as a function of gap, for a variety of vertical steering settings.

Canted Undulators

In the case of the canted undulator beamlines in sectors 21,23, and 24, the decision was made to incorporate the function of 2 (and a half) photon beam position monitors in the same vacuum housing, comprised of 10 blades (table 1). The electronics and signal processing is handled in a fashion quite similar to all of the other units. The radially outermost four blades, i.e. furthest from the center of the storage ring, are nominally centered on the upstream undulator's beamline and labeled "P1". The innermost four blades are centered on the downstream undulator's beamline and labeled "P2". The two blades in the center of the device are sensitive to photons generated by the small bending magnet between the two undulators, which is responsible for the 1 mrad angular separation between the two beams.

Incorporating so many blades in the same vacuum housing has the advantage of considerable savings in cost, since only a single vacuum enclosure, mechanical support structure, and translation stage is required vs. the other beamlines with have separate vacuum housings for P1 and P2. Given that the horizontal extent of the photon beam is given by K / γ, with K less than 3 or so and $1 / \gamma = mc^2/E = 73$ microradians, photons from the upstream device do not impinge upon the idxbpm blades aligned with the downstream device, and vice versa.

It was thought that this should have been adequate to decouple the response of one of the bpm channels from the "other guy's" gap. Unfortunately, if either of the gaps is below 15 mm or so, a cloud of photoelectrons is formed, which produces crosstalk between P1 and P2. A feedforward algorithm to compensate for this by applying blade offset corrections vs. the other guy's gap should reduce this effect, and preliminary investigations have been made, but much work remains.

Conclusion

A method for finding the "electrical" center of insertion device photon beam position monitors has been found which should reduce systematic position errors for most beamlines to better than +- 10 microns or so. Certain blade geometries result in unacceptably large errors and small signal levels. Work is proceeding on upgrading these assemblies.

References

Glenn Decker, Om Singh, "A Method for Reducing X-ray Background Signals from Insertion Device X-ray Beam Position Monitors," Phys. Rev. ST Accel. Beams 2, 112801 (1999).

G. Decker, O. Singh, H. Friedsam, J. Jones, M. Ramanathan, D. Shu, "Reduction of X-BPM Systematic Errors by Modification of Lattice in the APS Storage Ring," 1999 IEEE Particle Accelerator Conference, New York City, New York, March 29-April 2, 1999, Vol., 3, pp. 2051-2053 (1999).

Appendix A

Summary of machine studies shifts associated with insertion device photon beam position monitors during APS run 2004-02, i.e. the summer of 2004.

Shift
/home/helios/SR/daily/0406/21/3/gapScan

Separately logged blade signals using sddsmonitor while running SRCollectGapFFData

Typical result

cd /home/helios/SR/daily/0406/21/3/gapScan
plotSector 12

This is what got me started thinking about blade gains and
how to go about making the normalized curves more similar.

%%%

Shift
/home/helios/SR/daily/0406/23/1

Investigation of canted undulator P1/P2 crosstalk.
First modifications to gap scan data collection,
specifically the addition of file

/home/helios/SR/daily/0406/23/1/23ID/extraIDs.mon

to sddsexperiment input file, specifically

/home/helios/SR/daily/0406/23/1/23ID/23IDscan.exp

Results:
cd /home/helios/SR/daily/0406/23/1/23ID/

plot showing crosstalk effect

```
sddsplot -graphic=line,vary -legend=file -sep=5 -mode=y=offset \
23IDDS30mm2.sdds 23IDDS25mm.sdds 23IDDS20mm.sdds
23IDDS15mm.sdds 23IDDS11mm.sdds \
-col=ID23USAverageGap,ID23V"*tion1" \
-col=ID23USAverageGap,ID23H"*tion1" \
-col=ID23USAverageGap,ID23V"*tion2" \
-col=ID23USAverageGap,ID23H"*tion2"
```

plot showing improved performance from reestablishing P1 (US beam) blade
offsets at each DS gap setting

```
sddsplot -graphic=line,vary -legend=file -sep=5 -mode=y=offset \
23IDDS11mmC.sdds 23IDDS15mmC.sdds 23IDDS20mmC.sdds
23IDDS25mmC.sdds 23IDDS11mmC2.sdds \
-col=ID23USAverageGap,ID23V"*tion1" \
-col=ID23USAverageGap,ID23H"*tion1" \
```

-col=ID23USAverageGap,ID23V”*tion2” \
-col=ID23USAverageGap,ID23H”*tion2”

complementary plot showing effect of upstream device on P2, which
is looking at the downstream device primary beam. The blade
offsets were reestablished for each scan, i.e. the “C” in the
filename indicates that this compensation was done.

sddsplot -graphic=line,vary -legend=file -sep=4 -mode=y=offset \
23IDUS11mmC.sdds 23IDUS15mmC.sdds 23IDUS20mmC.sdds
23IDUS25mmC.sdds \
-col=ID23DSAverageGap,ID23V”*tion1” \
-col=ID23DSAverageGap,ID23H”*tion1” \
-col=ID23DSAverageGap,ID23V”*tion2” \
-col=ID23DSAverageGap,ID23H”*tion2”

variation of P2 (ds) blades with us device, ds device parked at 60 mm:

sddsplot -sep -graphic=line,vary -layout=2,2 \
-col=ID23USAverageGap,ID23”[ABCD]2:Corrected”
23DSBladesVsUsGap.sdds

and vice versa

sddsplot -sep -graphic=line,vary -layout=2,2 \
-col=ID23DSAverageGap,ID23”[ABCD]1:Corrected”
23USBladesVsDsGap.sdds

%%
Shift

/home/helios/SR/daily/0407/04/3

The curve fitting in its nearly final form was developed
to analyse data collected during this shift.

Second modification to gap scan procedure: edited the file

SR_msAve.mon to look at the msAve:[xy] versions of the raw nbbpm [BA]:P[01] pv's instead of the ErrorCC's. This was because I used controllaw to do simulataneous steering at all source points and I wanted the absolute nbbpm readbacks vs the error's which would only tell me how well controllaw was working.

Scripts:

The "final" version of the postprocessing script, in its most general form, dealing properly with all the unique nomenclature and blade geometries, is

/home/helios/SR/daily/0407/04/3/postProcessing/getOffsetsRev9

This script generates a lot of plots and printouts, which should be suppressed in the final product. I've annotated the script reasonably well so that you can tell what it is trying to do. It has been pretty well debugged.

The script performs two functionally distinct tasks:

1) For a single sector's idxbpms, uses sddsexpfit to fit individual blade signals to the form

F(gap) = Offset + Factor * Exp(-Rate * gap)

and record the fit parameters Offset, Factor and Rate in an output file, usually named e.g. ID07.proc

2) After subtracting the Offset and multiplying individual blade signals by Exp(+Rate * gap), compute corrected positions using appropriate delta / sum arithmetic, e.g.
Y2 = (A2 - B2) /(A2 + B2) etc., and compare with the case using only Offset subtraction.

An earlier version of this script, called simply getOffsets in the same directory works for most situations, just doesn't deal with as many special cases. I used it in later shifts sort of by mistake, but also because it runs quietly without all the plots and printing.

There is also a useful script

/home/helios/SR/daily/0407/04/3/postProcessing/makePutCommandsRev2

used to generate a cavput command that writes the fitted offset values to the OffsetP blade offset process variables for a given sector. This was used extensively during the later alignment shift 7/21/04.

For mass production, I wrote two scripts that use the getOffsets script repeatedly to do all the curve fitting for all the sectors. The script

/home/helios/SR/daily/0407/04/3/postProcessing/offsets/barfOffsets

fits all the sectors for a given data set, and combines all the fit coefficients into summary files combinedParams.sdds and combinedParamsGoodABCDOnly.sdds. barfOffsets is run from subdirectories e.g.

/home/helios/SR/daily/0407/04/3/postProcessing/offsets/x-0.5y0

where one output file per sector results, along with the summary combinedParams* files. The directory name x-0.5y0 corresponds to horizontal steering of -.5 mm at P1, but vertically approximately centered.

The script

/home/helios/SR/daily/0407/04/3/postProcessing/offsets/superbarf

runs barfOffsets repeatedly for all the datasets.

Results:

Final results at the very end of the data processing using the “best” fitting procedure after much trial and error, are in subdirectories of

/home/helios/SR/daily/0407/04/3/postProcessing/offsets

To see the results in all their glory,

```
cd /home/helios/SR/daily/0407/04/3/postProcessing/offsets
plotAll
plotAllOffset
```

I have scripts plotAllIndividually plotAllOffsetIndividually in this directory that use -layout=1,1 vs. -layout=2,2. This is to facilitate use of the mouse tracker during detailed alignment work which occurred later in the run.

To interpret the plots, curves labeled with the suffix “raw” show computed position using only blade offset compensation, with the offsets derived from the fit. The position gain is effectively set to unity, so these positions are in delta over sum units. For example

define A = S5ID:P1:A:CorrectedM - fitted P1A blade offset
etc. for B, C, D, then

Then

ID05X1raw = (A + B - C - D) / (A + B + C + D)

The plots without the “raw” suffix use both blade offset subtraction and “blade gain” compensation, i.e. substitute

A = (S5ID:P1:A:CorrectedM - fitted P1A blade offset)* exp(rate*gap)

etc. in the delta/sum formula to get ID05X1, where “rate” is the fitted rate for S5ID:P1:A in this case.

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
Shift

/home/helios/SR/daily/0407/20/1

This was a followup study to the shift

/home/helios/SR/daily/0406/23/1

attempting to perform detailed alignment in order to reduce gap dependent systematics for the canted undulator beamlines.

I quote:

“It appears that my method for aligning the xbpm’s is an abysmal failure at 23ID.”

It wasn’t quite as bad as all that, but much work remains on the canted undulators.

%%%%%%%%%%%%

Shift

/home/helios/SR/daily/0407/21/1/xbpmAlignment

Using the data collected during the july 4th shift, the xbpm’s were physically moved closer to the sweet spots, which ultimately will result in reduced position readback variation with gap. The two translation stage controllers played leapfrog around the ring (Mike

Hahne did the leaping) until all idxbpm's in sectors

5,6,7,8,9,10,12,13,15,16,17,18,19,20,22,31,32,33

had been visited and aligned.

Scripts:

I notice a softlink getOffsetsRev9 in this directory, linking back to the script of the same name from july 4th. This is the post processing script.

There is also a softlink makePutCommandsRev2 back to the july 4 script of the same name. Output from this script can be seen for example in files

/home/helios/SR/daily/0407/21/1/xbpmAlignment/ID05/put05Offsets-001

The scripts

/home/helios/SR/daily/0407/21/1/xbpmAlignment/makeExpFile
/home/helios/SR/daily/0407/21/1/xbpmAlignment/makeBriefExpFile

do what they say, writing a properly formatted sddsexperiment input file to stdout. The only difference between them is that one uses the list of gap values in file gapValues.sdds while the other uses an abbreviated list in file gapValuesBrief.sdds. The brief version runs faster and only over the range from 20 to 35 mm, which is all that is needed to extract blade offsets using sddsexpfit.

The scripts

/home/helios/SR/daily/0407/21/1/xbpmAlignment/plotLookupTableComparison
/home/helios/SR/daily/0407/21/1/xbpmAlignment/plotLookupTableComparisonOffset

compare lookup tables before and after the alignment, and essentially
show the final results of the shift.

A generalization of the above scripts named

/home/helios/SR/daily/0407/21/1/xbpmAlignment/plotLookupTables

allows comparison of any two installed lookup tables, i.e.
those archived in files matching

/home/helios/oagData/feedForwardFiles/IDgap/FF.*

We really need something like this available as a button on
SRCollectGapFFData to compare recently collected data with archived
data.

Results:

cd /home/helios/SR/daily/0407/21/1/xbpmAlignment
plotLookupTableComparison
plotLookupTableComparisonOffset

%%%%%%%%%%%%%%%%%%%%%%%%%%%%5

Shift

/home/helios/SR/daily/0408/02/3

This was essentially a repeat of the massive data collection
performed july 4th, but with a finer scale and more curves
(7 vs. 5). The objective was to see how much I improved
things during the alignment fest july 21.

I was a bit rushed setting up for this shift in comparison
to the careful thought that went into the 7/21 shift. For

example I reverted back to the older getOffsets script, instead
of the newer getOffsetsRev9. The fits are the same in either
case for the sectors under study though.

Scripts:

I updated the Steer* scripts that work with the square controllaw
matrices to effect the simo-angle bumps, to steer in 100 micron steps
at the P1 location vs 250 microns that was used july 4. They are just
cavput commands to nbbpm P0 setpoints.

I had to split the script plotAll used july 4 in half since unix
balked at such a large command for seven vs. five families of curves.

Results:

cd /home/helios/SR/daily/0408/02/3/postProcessing
plotAllX
plotAllXOffset
plotAllY
plotAllYOffset

Pretty good in most cases. The green (or center-most) trace is close
to the ubop steering. A few shutters were closed which was annoying,
and several sectors were still pretty badly misaligned. Data from
this shift was used the last day of the run 8/26 for one last
marathon xbpm alignment shift.

%%%%%%%%%%%%%%%%%%%%%%%%%%%%

Shift

/home/helios/SR/daily/0408/26/2/xbpms/gapScans

A repeat of the 7/21 translation stage alignment shift, using the data
from August 2. I was able to get a three-curve family of gap scans in

each plane in one hour, to QA the final alignment picture.

Scripts:

I was also using the silent-running getOffsets script here.

The plotAll* scripts had to be edited to use 3 vs. 7 curve families.

Results:

```
cd /home/helios/SR/daily/0408/26/2/xbpms/gapScans/postProcessing
plotAllX
plotAllXOffset
plotAllY
plotAllYOffset
```

This shows that I'm very close to optimal alignment in many of the sectors.

Appendix B

Insertion Device Photon BPM Blade Gain
Software Functional Requirement
(sent to ASD-CTL September 2004)

As a result of extensive machine studies during run 2004-02, the inclusion of a "blade gain" correction to insertion device photon bpm signals should make it possible to reduce gap-dependent systematic errors to below the +- 10 micron level.

Empirically, individual blade signals have been found to fit quite well to a function of the form

F(gap) = Offset + Factor * Exp(-Rate * gap)

Unfortunately, different blades from the same bpm exhibit different

“Rates”. (I am using the terminology from the program sddsexpfit, which I use for much of the fitting process).

By using a special alignment procedure I have been able to displace the bpm relative to the user’s desired beamline such that the Rates for any given bpm are comparable, or at least not vastly different. Even so, residual systematics after the alignment are as high as 80 microns so for many units.

All three coefficients Offset, Factor, and Rate are strong functions of the relative alignment between the beam and the bpm. For the Offset, the physics is clear, since it is simply a measure of the open-gap residual radiation field. The variation of the”Factor” is in essence what makes the bpm a bpm. I will confess to being somewhat mystified as to the physics of the variation of Rate with position, however I don’t need to understand it to be able to compensate for it.

So. At present, taking S17ID:P1:A as an example, we have

S17ID:P1:A:CorrectedM = S17ID:P1:A:NormAveSU * (100 / S35DCCT:currentCC)

S17ID:P1:A:AdjustedM = S17ID:P1:A:CorrectedM - S17ID:P1:A:OffsetP (i.e. the blade offset)

Defining A = S17ID:P1:A:AdjustedM, and similarly for B, C, and D,

S17ID:P1:ms:x = S17ID:P1:ms:x:GainAO * (A + B - C - D) / (A + B + C + D)

S17ID:P1:ms:x:AdjustedCC = S17ID:P1:ms:x - S17ID:P1:ms:x:OffsetAO

S17ID:P1:ms:x:ErrorCC = S17ID:P1:ms:x:AdjustedCC - S17ID:P1:ms:x:SetpointAO

etc.

My understanding is that all of this computation takes place at
1.5 kHz on a DSP board located in the local feedback ioc crate.

(Note from Frank Lenkszus' response:
It's a little more complicated. Two calculations are done in parallel. The DSP does a caculation through the equivalent of []; AdjustedM and computes the delta/sum position and deposits that in reflective memory for the datapool. This is done at 1534 Hz. The DSP passess the Blade currents to the IOC processor which does precisely the caculations as you describe. So, the blade gain factor would need to be applied to both calculations, the DSPs and IOCs.)

My proposal is to apply a multiplying factor (e.g. S17ID:P1:A:GainP) to the AdjustedM value at the feedback clock rate. This factor should be a calc record derived (at a much lower speed) from the associated insertion device gap, and equal to e.g. Exp(+Rate * ID17:Gap). The quantity "Rate" I propose to be a new process variable, changed periodically as part of the feedforward lookup table generation process involving gap scans. I will be putting together a separate specification for the gap scan tcl script.
Both the blade offsets and Rates will come out of the scan. The individual blade Rates should be added to the SR save compare restore database. One complication is the inconsistent naming conventions for the insertion device gap readback process variables around the ring (see footnote below).

Given the strong dependence on steering, most of the same operational rules will apply, in particular that idxbpms must be removed from controllaw following beamline steering. A new scan will be required after any steering, with the associated installation of new offsets and rates for the affected photon bpm's. It remains to be seen whether gap feedforward can be discontinued altogether.

So my question to Frank is whether or not this is possible, and would it be possible to try this for the October startup. Initially the multiplying factors could default to unity, which shouldn't change functionality in any material way.

any comments are welcome

thanks GD

PS-

ID gap naming (non)conventions:

“most” sectors use ID??:Gap, where ?? is a two digit sector designation, i.e. 07 and not just 7, for sectors less than 10.

This mostly standard convention is used in sectors

1, 4-13, 15-20, 22, 31-34

The canted undulator straight section uses a better nomenclature, which I think should be applied to all sectors, even those with only a single device installed.

ID23ds:Gap and ID23us:Gap

correspond to the downstream and upstream devices respectively.

This us/ds nomenclature is used in sectors 2, 14 (ds only), 21 (us only), 23, and 24.

Then there are the oddballs:

ID03:Gap is the downstream device
ID03us:Gap is upstream

The cpu in sector 4 is hopeless for the xbpm’s, as is sector 11 emw, although the emw has a mechanical gap control that follows the convention as described above.

In general I don’t have a solution for sectors with two devices (sectors 2,3,4, and canted sectors 21,23,24), although the canted undulators are very close to being solved. Sector 2 has been clamoring for idxbpm support, but a 2-dimensional lookup table is pretty ungainly.

Appendix C

Request for new tcl application
(sent to OAG group)
Suggested name = SRFitGapScan

The desired new functionality is very similar to a lot of what the script SRCollectGapFFData already does. There are enough unique new features that I feel a completely new application will be needed, one which will use much of the existing code from SRCollectGapFFData .

Overview:

The functions of this new application are

1) To scan a set of insertion device gaps, almost exactly as is presently done by SRCollectGapFFData. The main differences will be that fewer gap values are to be used, and the “Corrected” blade signals e.g. S7ID:P1:A:CorrectedM must be logged in addition to what is already being logged. Also, the nbbpm ErrorCC’s are logged by SRCollectGapFFData, but I would prefer logging the msAve:[xy] pv’s instead.

2) To use sddsexpfit to fit these corrected blade signals
to a function of the form

F[gap] = Constant + Factor * exp[- Rate * gap]

3) To download the “Constant” fit parameters to existing blade offset process variables of the type e.g. S7ID:P1:A:OffsetP.

4) Later, following the addition of new epics pv’s that Frank is working on, to download the “Constant” fit parameters to the OffsetP pv’s in addition to downloading the “Rate” process variables to not-yet-existent pv’s, most likely of the form S7ID:P1:A:RateP.
I have made a special request to Frank to use this nomenclature.

Specification

Function 1)

a) Add the file

/home/helios/SR/daily/0407/04/3/IDcorrectedPVs.mon

to a new &measurement_file entry in the sddsexperiment .exp input file.

b) Use the file

/home/helios6/SR/daily/0407/04/3/SR_msAve.mon

which uses nbbpm msAve:[xy] pv’s in place of the present ErrorCC pv’s, presently located in the file

/home/helios/oagData/sr/gapScans/feedforwardInputFiles/SR_msAve.mon

There are also id xbpm msAve:[xy] values in both files.

c) Use the gap values listed in the file

/home/helios/SR/daily/0407/21/1/xbpmAlignment/gapValuesBrief.sdds

in place of the complicated way that this is done by SRCollectGapFFData . The list of gap values is the same for all devices, namely 20, 22.5, 25, 27.5, 30, 32.5, and 35 mm.

This means that many of the tcl variables used by SRCollectGapFFData are no longer needed, specifically

Maximum gap value
Maximum step value
Minimum gap value
Minimum step value
Number of same size steps
Factor for a step increase
Number of points in scan
Delta to minimum gap

The step counter display is also not needed.

d) Buttons:

- The Run button should essentially do the same thing as for SRCollectGapFFData

- The Goto Minimum Gap button should be replaced with a Goto 20 mm Gap button, i.e. the first point in the scan.

- We don't need the preview button

- The Review Scan... button should be used to display the raw corrected blade data, using symbols since there are only 7 data points.

%%%%%%%%%%%%%%%%%

Function 2)

Fitting the data. There should be a "Fit Data" button in place of the "Prepare FF Data" button.

Pushing this button should in effect execute the script

/home/helios/SR/daily/0407/04/3/postProcessing/getOffsetsRev10

one time for each device scanned, followed by sddscombine -ing the resulting output files, using the -merge option.

Supposing the gap scan data is contained in file scan.sdds, then executing the command e.g.

```
getOffsetsRev10 07 scan.sdds ID07.proc
```

will perform the fit for sector 7 and place the fit parameters, original data, fitted data and residuals into the output file named ID07.proc .

Once all of the fits for the different sectors are complete, the command e.g.

```
sddscombine -merge ID07.proc ID08.proc allData.proc
```

will place all of the fit parameters and columns into a single output file, in this case using data from just sectors 7 and 8.

There are some intermediate files which are created and later deleted, which I'm sure I did wrong. They should go into some tmp directory with unique filenames I suppose.

There is a lot of logic separating out all of the different cases associated with all the different ways that insertion device gaps process variables are labeled, and dealing with the different idxbpm blade geometries. See my viewgraphs located on page 10 of the file

http://www.aps4.anl.gov/diagnostics/viewgraphs/icalepcs01/landscape-Viewgraphs.pdf to see the difference between “ABCD” and “ABEF” geometry. This matters when it comes time to perform the delta/sum arithmetic, but is simply a nuisance here. The canted undulators add a couple more twists in the logic.

%%%%%%%%%%%%%%%%%%%

Function 3)

Download the fitted offsets to process variables. This should be attached to a button labeled “load offsets”. During studies I wrote a silly script that writes a series of cavput commands to stdout, to be found at

/home/helios/SR/daily/0407/04/3/postProcessing/makePutCommandsRev3
.

Supposing the fitting routine produces an output file named allData.proc , the command

```
makePutCommandsRev3 07 allData.proc > loadOffsets
```

will result in a file loadOffsets having contents e.g.

```
#! /bin/csh -f
cavput -list=S7ID:P1:A:OffsetP=6.676770579197600e-01 -pend=5
cavput -list=S7ID:P1:B:OffsetP=5.203255567431273e-01 -pend=5
cavput -list=S7ID:P1:C:OffsetP=4.362981866849583e-01 -pend=5
cavput -list=S7ID:P1:D:OffsetP=4.199530242424707e-01 -pend=5
cavput -list=S7ID:P2:A:OffsetP=-2.008569867399404e+00 -pend=5
cavput -list=S7ID:P2:B:OffsetP=-2.531978530215982e+00 -pend=5
cavput -list=S7ID:P2:E:OffsetP=8.852830545245965e+00 -pend=5
cavput -list=S7ID:P2:F:OffsetP=1.151521084618385e+01 -pend=5
```

where the numerical values come from the appropriately-named parameters in allData.proc, in this case ID07A1Const, ID07B1Const

etc.
The commands

chmod +x loadOffsets
loadOffsets

actually execute the cavput commands.

There is a lot of the same logic sorting out all of the special cases here as well, with the additional complication that the pv's S7ID:P1:A:OffsetP etc. do not have a prepended zero associated with the sector number as do the id gap pv's.

There is most likely a much easier way to do this.

%%%%%%%%%%%%%%%%%%

Function 4)

Another button labeled "load offsets & rates" should perform the same function as Function 3) above, in addition to loading parameters e.g. ID07A1Rate to pv's S7ID:P1:A:RateP etc.

%%%%%%%%%%%%%%%%%%%%%%%

I think that's it.   GD